

NOVEMBER 1984

DCIEM No. 84-R-45

**A SPATIAL DATABASE MANAGEMENT SYSTEM
FOR COMMAND AND CONTROL ***

**C.A. McCann
M.M. Taylor
M.I. Tuori**

*** Reprint of a paper presented at the International Conference on Advances in Command Control and Communications Systems Theory and Applications, Bournemouth International Center: 16 - 18 April, 1985**

**Defence and Civil Institute of Environmental Medicine
1133 Sheppard Ave. West, P.O. 2000
Downsview, Ontario M3M 3B9**

DEPARTMENT OF NATIONAL DEFENCE - CANADA

TABLE OF CONTENTS

ABSTRACT	1
INTRODUCTION	2
BACKGROUND	2
Human Processing of Spatial Information	2
ISIS CAPABILITIES	3
SDBMS-1: A DEMONSTRATION TESTBED	4
Information Modelling	4
Database Design	6
User Interface	6
Graphical Interaction	7
Base Pictures	7
MASCOT Development Environment	8
MASCOT Design of SDBMS-1	9
Implementation	10
FUTURE WORK	10
User Dialogue Design	10
Tools for the Tactical Planner	11
Spatial Data Representation	12
Time-dependent Data	12
REFERENCES	13

ABSTRACT

Methods for structuring, storing and presenting spatial information have not been given enough emphasis in the development of current command and control systems. This paper will describe the Interactive Spatial Information System (ISIS) project, whose purpose is both to study the interaction of humans with spatial data for decision aiding and to provide models on which the operational spatial information systems can be based. The paper will outline the motivation behind the project, characteristics of an ideal ISIS, and our first implementation of a working spatial database management system that demonstrates some of these capabilities.

INTRODUCTION

The Interactive Spatial Information System (ISIS) project at DCIEM has a double purpose: to study the interaction of humans with spatial (map-like) data for decision making, and to provide models on which operational spatial information systems can be based. The models take the form of working systems, whose design derives from the conceptual understanding achieved at some particular stage.

BACKGROUND

Human Processing of Spatial Information

We are concerned that the presentation and manipulation of spatial information has not been given enough emphasis in the development of current command and control systems. Psychological studies reviewed by Taylor and Taylor (1) have shown that humans manipulate information both logically (linguistically) and associatively (spatially). The spatial mode of thought enables humans to connect different aspects of patterns into a coherent whole, to notice consistencies, to recognize analogies, to "see the big picture." It is the intuitive half of thought. The logical mode, however, permits the construction of sequences, fine discriminations, interrelation of multiple contingencies, and the extraction of correct interpretations from a multitude of possibilities. The logical mode is much better than the spatial in analyzing the details of complex situations. Spatial display helps the human to take advantage of the intuitive mode of thought, whereas language (including the language of mathematics) gives preference to the logical mode. The two modes are complementary, and together provide a power that neither alone can achieve.

Spatial arrangements of military information allow the presentation of easily recognized structural patterns (e.g., military groupings suited for the functions of attack, mutual support, area defence, etc.). Since space is continuous, spatial displays permit presentation of dense information structures in which neighbouring information is highly correlated. A picture of an area in which manoeuvres are anticipated can often be much more informative than a description. The combination of picture, map, and description can together give a much better idea of the possibilities and problems than can any single mode of display.

It is important to recognize that spatial display of data and the display of spatial data are not the same. Spatial data are those whose relationships depend on their location rather than on their function. The control of a military operation is a good example of a situation where the display of spatial information (positions of friendly and enemy military units and resources with respect to the terrain) is very important. Spatial display of non-spatial information is also possible and often useful. For example, the "Media Room" developed by Bolt (2) at the Massachusetts Institute of Technology (MIT) displays all sorts of information laid out spatially so that the user can "move around" in it, plunging deeper into items of interest, or scanning the display space for useful material. Some of the information is spatial, as when the user dives into a map of MIT to focus on a single building, of which an aerial picture may be displayed, and then into a single office in that building, then into a book lying on a desk in that office. Other information, such as a calendar of events, is non-spatial, but it can be found from spatial memory because it is visually located at a particular place in the large display

space.

The desirability of having command and control systems that process, manipulate and display spatial information to the military user is obvious. To our knowledge, no system for doing this exists.

The development of our ideas for an Interactive Spatial Information System has been coloured by familiarity with military land operations, and, in particular, with a concern for some of the problems of tactical planning, where spatial information is used extensively (McCann and Moogk (3)). However, the concepts described in the following sections can be applied to all command and control environments.

ISIS CAPABILITIES

At a minimum, ISIS should provide the military user with a highly interactive environment which will allow the user to scroll freely on a very large virtual map; zoom in on particular areas of interest; call up and remove spatial information on his display; modify stored information; and query the system for answers to particular questions or to access geographic problem solving tools. For example,

- a) It should be able to compute the answer to direct questions such as "How far is it from point A to point B using routes invisible from position C."
- b) It should serve as a repository of information such as "There exists a class A road named Highway 262 that follows these coordinates" or "Smoke has been reported from Hill X."
- c) It should, on request, display information about regions, about types of entities, about battle situations.
- d) From spatial information it contains, it should be able to synthesise or select visual imagery showing what certain places would look like from certain viewpoints.
- e) It should be able to compute mobility conditions for selected areas, taking into account vegetation, ground conditions, slope and recent weather.

Much more becomes possible with the inclusion of various aspects of artificial intelligence, such as a Natural Language Interface (NLI); expertise in logistical requirements, military doctrine, and so forth; and particularly intelligence in the system's approach to dialogue:

- a) If an item of information creates an apparent inconsistency (e.g. a position for a tactical unit is incompatible with previous information), the system should be able to request confirmation, giving reasons for its request and attempting to discover whether the difficulty is in the new information or in some aspect of the old.
- b) The system should keep track of both its interlocutors and their responsibilities, so that it can respond appropriately to implied questions. The mode of response should be appropriate to both the enquirer and the query.
- c) The system should have some knowledge of the rules of military engagement, so that it can project the likely outcomes of suggested operations.

SDBMS-1: A DEMONSTRATION TESTBED

The Spatial Database Management System (Version 1) is a demonstration testbed in which some of the ideas of ISIS have been implemented. It is intended both as a vehicle to demonstrate concepts and as an environment for conducting experiments on the human factors aspects of interaction with spatial information.

A previous investigation of the field of spatial data handling (Moon et al (4)) had concluded that the relational approach to data modelling and structuring was the most appropriate of the currently available database management techniques. To reduce implementation time, a commercially-available relational database management system (DBMS) called Mistress (5) was used for the core of SDBMS-1.

Two extensions to Mistress were required to permit handling of spatial data. The first was the provision of a variable length bulk data type of arbitrary format. Bulk data in SDBMS-1 include X,Y coordinate streams, raster imagery (e.g., Landsat) or strip-trees. A mechanism was also required to allow special spatial operators (e.g., distance, area, containment) to be used in the high level query accesses to Mistress. Mistress was therefore modified to allow "external operators" which are defined by the application. This modification was seen as essential for the efficient incorporation of geographic calculations such as distances or more complex tactical planning aids such as intervisibility.

Information Modelling

The information in SDBMS-1 was conceptually structured using a top-down approach called Entity-Relationship (ER) modelling. In brief, the ER approach models information from the end user's point of view by specifying the classes of objects and the significant relationships that exist amongst the classes. The entities and their relationships are conveniently depicted as a diagram, called an ER diagram, in which a rectangle is used to represent each class of object and a line joining two classes indicates a relationship between them.

For tactical planning, object classes in the database of SDBMS-1 needed to include topographic features such as roads, rivers, towns, and lakes; and tactical features such as friendly or enemy units, equipment, boundaries, control measures and obstacles. A subset of the ER diagram for tactical information is shown in Figure 1. The relationships among the object classes link them together in a network-like structure. For example, the relationship *owns* might be used to describe the ownership of military equipment by friendly units. Weapons can be further linked to the class *ammo* by a relationship called *uses_ammo*.

Once the objects and their relationships have been at least partially defined, each can be examined to determine the characteristics or attributes that will be needed to describe each instance of that object. The object *road* might require attributes that describe its type (e.g., expressway, city street), width, and name, as well as its X,Y location. Attributes for *inf_wpn* (Infantry weapon) might include name, type, weight and range. Included in the attributes for a relationship are those that provide the linkage between the objects they join; in the case of *owns* the attributes must include the name of the owner and the equipment name and class. Equipment class is necessary in this case, since *owns* links units with several classes of equipment (e.g., Infantry weapon, vehicle).

Figure 1. Entity-Relationship diagram for tactical information

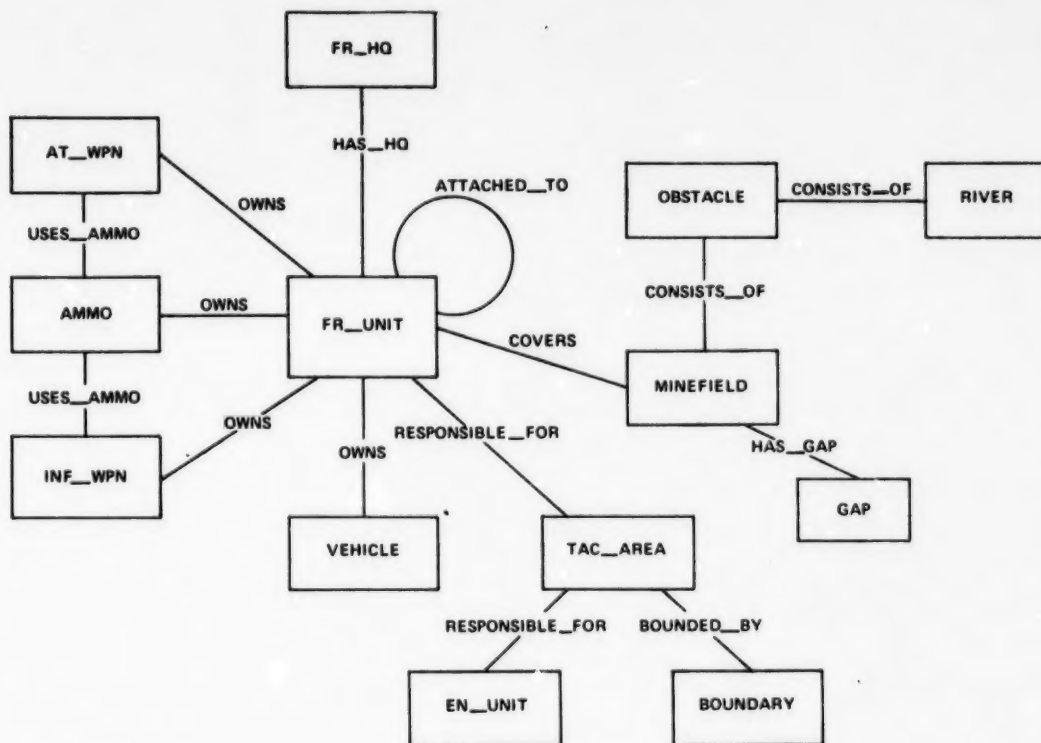


Table 1. Friendly unit (*fr_unit*) table for SDBMS-1

name	parent	role	size	no_pers	dtg	x	y
10LLAD	1CDN	arty	regt	547	240530	300000	4884000
11CMB	1CDN	mechInf	bde	5496	232350	308000	4892000
11RIFLE	1RRC	Inf	coy	129	240600	324300	4886500
12CMB	1CDN	mechInf	bde	5185			
12RIFLE	1RRC	Inf	coy	827	240600	331000	4891000
13CLH	13CMB	armd	regt	689	240530	321300	4882500
13RECCE	13CMB	recce	sqn	140	240200	322500	4880500
162MOR	16CBTSP	mor	plt	31	232400	331900	4888300
16CBTSP	1RRC	sp	coy	159	232000	317000	4885500
17RIFLE	1RHLI	Inf	coy	121	240600	325300	4876600
1RRC	13CMB	Inf	bn	827	240630	325500	4889000
951LLGUN	95LLGUN	arty	tp	28	240400	302300	4885300

Database Design

Upon completion of the conceptual data model, the next stage, database design and construction, is straightforward. Each object or relationship becomes a table in the relational database, and the attributes become the column headings in the table. (Part of the friendly unit table, *fr_unit*, is in Table 1.) When the database is actually filled with data, each row in the table contains one unique instance of the object, and is assigned a unique identifier. In addition to the X,Y coordinates of the object's position, the attribute list includes the co-ordinates of the minimum, horizontally oriented, rectangle into which the graphical representation of the feature fits. The use of the minimum enclosing rectangle (MER) in addition to the actual X,Y coordinate representation of the feature permits many spatial calculations such as picking a feature from the graphics screen, to be carried out more efficiently.

Data modelling and design is not a trivial job. The efficient design of the ER network has strong bearing on the later efficiency of the whole system, both in terms of how easy it is to ask questions and how fast the system can answer them. The use of the relational model does, however, facilitate extension of the database by the simple addition of new relational tables. The development of the topographic model used with SDBMS-1 and the value of the ER approach is discussed in more detail in Tuori and Moon (6).

User Interface

An interface that allows both semantic and graphical interaction is an important aspect of the design of SDBMS-1. The user has the option of input through keyboard, voice and gesture, while information is output on either an alphanumeric or a graphics display screen. The user formulates system requests semantically by typing them on a keyboard or by speaking them through a voice recognition system. In many cases, query formulation will require that the user refer to graphical information already displayed (e.g., to indicate an area of interest, or to point to a displayed feature) or input new graphical information (e.g., draw a new feature). Graphical interaction of this sort is done using a pointing device (in the current version, a mouse).

Because SDBMS-1 is based on a conventional DBMS, interactions with it are constrained by the formal query language of the DBMS (slightly modified). There are ten types of commands. Each is initiated by a keyword that may need to be followed by further inputs to complete the command. A brief description of each command follows:

- show - extracts and displays information about one class of object in the database
- unshow - erases information from the display
- join - extracts information concerning more than one class in the database
- move window - changes the geographic area covered by the display
- insert into - allows a new object to be added to the database
- update - changes the values of attributes of an existing object in the database
- remove from db - removes objects from the database
- sketch overlay - allows the user to annotate the graphic display
- base pic and history - allows manipulation of the internal base picture and history
- quit - ends the interactive session

There are two main categories of query: those affecting the information currently displayed on the graphics and text output screens; and those affecting the information

stored in the database. Formulation of any command to access the database requires the specification of the appropriate keyword, followed by the class(es) of object concerned, the aspects or attributes of the object and any constraints on the selection of instances from the class (given as a Boolean expression). The syntax is of the following form:

<keyword> <class> aspects <attribute> such that <Boolean expression>;

In a spoken query, the final semicolon is replaced by "please", and commas are spoken as "comma". The spoken command to retrieve the role and size of a particular friendly unit would be:

show fr_unit aspects role comma size such that name match '1RRC' please

Graphical Interaction

SDBMS-1 allows the user to integrate graphical gesture into a query. If the friendly unit named 1RRC was already displayed on the graphics screen, the user could refer to it (using the keyword "this") as an example of the class of object required in the following query:

show this aspects role, size;

To complete the query, the user picks, by graphical gesture, the unit named 1RRC from the graphics screen. The system deduces that the selected object is of class *fr_unit*, substitutes that fact in the query and responds, giving the user the role and size of the other friendly units.

The graphic description of areas can also be used as a constraint in a query. The following query will show only those friendly reconnaissance units within a geographic area described graphically by the user:

show fr_unit aspects X,Y such that role match 'recce' and x,y within here;

Graphical interaction is also essential when the graphical attributes of an object (such as its location) must be inserted or modified in the database. To change the boundary associated with a friendly Brigade, the user would pose this command:

update boundary such that this fill in aspects coord please

The last part of the query tells the system which attribute is to be updated (i.e., *coord*, or location). The user is then placed in a graphical editor which provides an environment for drawing the new boundary.

A more complete functional description of the user interface to SDBMS-1 can be found in McCann et al (7).

Base Pictures

Part of the information contained in a spatial database for tactical planning can be considered to be relatively unchanging — primarily topographic features such as roads, rivers, vegetated areas, etc. It is usually desirable to display at least some features of this sort to provide a backdrop for the more transient tactical features for command and control, namely military units, boundaries, etc. SDBMS-1 provides a mechanism for the retrieval of *base pictures* — raster images which have been previously created from the database using "show" commands. It is usual to start an interactive session with a display of a base picture. Each query to the database then results in a single overlay of information, analogous to a drawing made on a transparent sheet lying over the base picture.

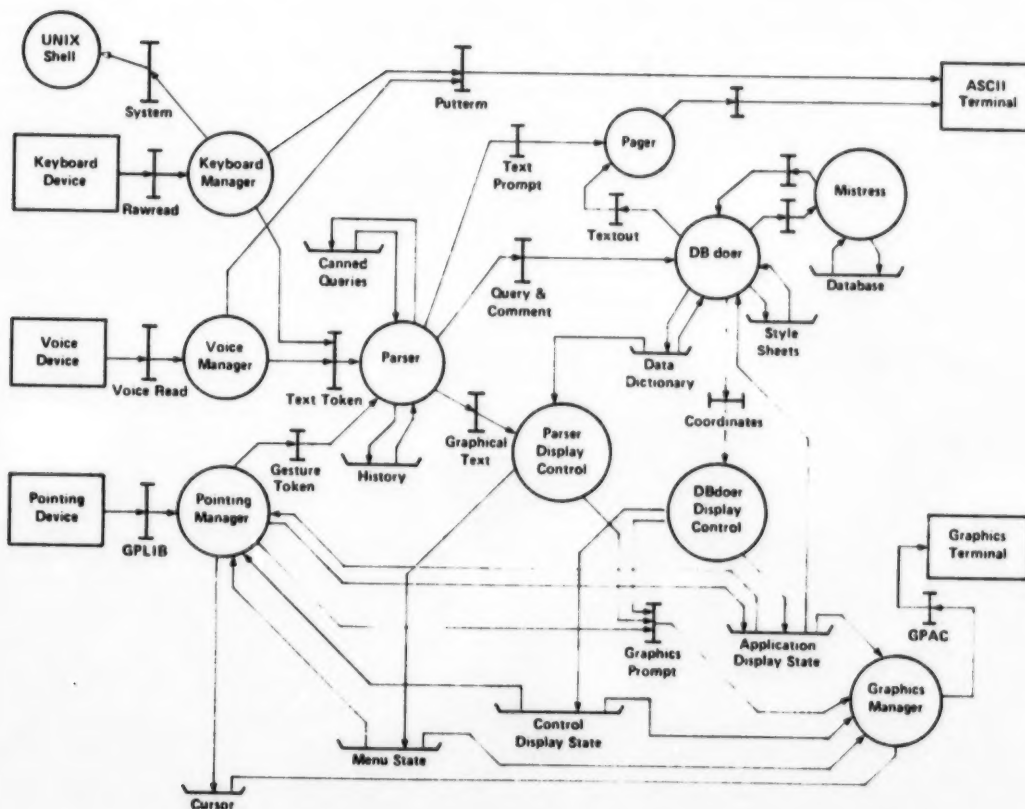
One advantage of employing base pictures is that the time-consuming queries required to generate the picture are circumvented. However, the objects visible in the base picture, being simply raster images, cannot be referenced, whereas objects generated

by a show command can. To emphasize this difference, the base picture should be displayed at lower contrast than the overlays. Each overlay contains not only the information on the display screens, but also includes a stored historical record of the query that was used. If the area of interest is changed by panning or zooming, the query can be run again either automatically or by specific user request, with reference to things within the new boundaries.

MASCOT Development Environment

Because SDBMS-1 is a testbed, its design had to cater for fast and easy modification. A design and implementation approach called MASCOT was selected as the environment most likely to ensure this. MASCOT stands for Modular Approach to Software Construction Operation and Test; it was developed at the Royal Signals and Radar Establishment (UK) in the early 70's (Jackson (8); Jackson and Simpson (9)). In addition to providing a design approach, as described below, MASCOT allows real-time concurrent processing, either on a single computer or over a distributed network of computers. At the moment, SDBMS-1 is implemented on a single processor whose UNIX operating system has been modified to support MASCOT.

Figure 2. MASCOT diagram for SDBMS-1



In order to discuss the MASCOT design of SDBMS-1, it is necessary to briefly define the MASCOT notation and conventions. A MASCOT design can contain three kinds of entities. The first, the Activity, is simply a set of sequential code. Individual Activities communicate with each other only through common data areas called Channels and Pools. Channels pass messages among Activities; once a message has been read by an Activity, it is gone, and cannot be accessed again. Pools, in contrast, hold information that needs to be read by several different Activities or several times by the same Activity. The information in a Pool is stable until it is deliberately changed by an Activity. MASCOT Activities, Channels and Pools are connected together in an "ACP" diagram, a network which specifies all the legal means by which information can flow between the Activities.

MASCOT Design of SDBMS-1

The MASCOT diagram for SDBMS-1 is in Figure 2. Activities in the diagram are represented by circles, Channels by I-beams and Pools by buckets. A fourth type of entity, the Device (represented by a square) is also used in this diagram. The purpose of the Device is to connect the net to the outside world. The general flow of information in this diagram is from left to right; input devices are represented on the left side of the diagram, and output devices on the right. Activities on the top are concerned with processing of semantic or textual information, while those on the bottom deal with the graphical.

The Parser activity is responsible for receiving, through the Text Token and Gesture Token channels, all input into SDBMS. Text streams are sent by the keyboard and voice devices, and gestures by the pointing device. Input for a single query can be intermixed in the three input streams as desired by the user. Furthermore, the input streams are asynchronous from the point of view of the human. A graphical gesture that should, syntactically, be placed at the end of a query being formulated on the voice device can be input at any time using the pointing device. It is the Parser's job to find the right place for it in the query and transform the text and gesture tokens into a standard format for passage to the DBdoer. Provision for this multimodal interaction is one of the important features of SDBMS-1.

The DBdoer activity interprets requests to the Mistress DBMS, which in turn accesses the Database pool for retrieval and storage of the spatial information in the relational tables. DBdoer is also responsible for determining where the retrieved information will go — whether to the ASCII Terminal device or to the high resolution Graphics Terminal. It does this on the basis of information on the *style* of display, contained in the Style Sheet pool. Style sheets specify the default display mode (text or graphics) for all attributes of all objects, and also describe the coding method to be used (e.g., for graphical attributes, possible coding methods include colour, size, shape, texture, etc.). The user can, to some degree, specify the style of the displayed information. For example, by inserting the phrase "as graphic" in a query, thus:

show frunlt aspects name as graphic please

the user can force the name of the unit, normally displayed on the alphanumeric screen, to be displayed as a graphic label. Certain dependencies are incorporated in the descriptions of style. For example, it makes no sense to display unit names on a map unless the unit symbol is displayed. Hence a request to show name as graphic will result also in display of the associated symbol.

Further control of text output is provided by the Pager activity. Management of graphics output is somewhat more complex, as reflected by the several pools at the bottom of the diagram. The system must keep track of the objects on the graphical overlays (via the Application Display State pool) and provide a tight coupling to a graphics

cursor for fast feedback to the user on the position of the pointing device.

Although not implemented yet, it is our intention to provide a menu-based interface for query formulation. The Parser Display Control activity will handle interaction with the menu, with the menu currently being displayed to the user stored in the Menu State pool.

Implementation

SDBMS-1 has been partially implemented on a Perkin Elmer 3242 computer and MASCOT under UNIX V7. Digitized terrain data for a small part of southern Ontario (an area roughly 80 x 30 km.), corresponding to that available on Canadian 1:50,000 topo maps, has been loaded into the database. A small amount of fictitious tactical data has also been stored. The current size of the database is about 31 Mbytes.

Our experience using MASCOT in this implementation has been positive. A small part of the MASCOT net (the text-handling part) was implemented initially as this was our first experience using MASCOT in a project of this size. Although work proceeded at that stage more slowly than we expected, the addition of other modules later on was relatively fast and easy. Since the interaction of these additional modules with the already-implemented part of the system was well-defined (being only through the Channels and Pools), the extensions had no ill effect on the existing code.

FUTURE WORK

User Dialogue Design

The main emphasis in our future ISIS work will be on upgrading the user interface. The interface currently provided in SDBMS-1 requires that queries be formulated precisely; the user must know the syntax of the query language and, as well, be very familiar with the structure of the information model. For example, to pose a query about the number of antitank weapons of a given type held by friendly units, the user must know the names of the tables (*fr_unit*, *owns*, *at_wpn*), the attributes of the tables that are pertinent to the query, and how to link the three tables, using the "such that" clause. The query is posed primarily by textual means.

One way of assisting the user with the syntax of queries would be to provide a menu of options (e.g., keywords, object classes, attributes) at each stage in query formulation. The MASCOT design of SDBMS-1 has provided for this. Rather than implement menu handling capabilities on older hardware that may not be able to provide a sufficiently fast response, we have decided to use an Apple Macintosh as the basis for a user front-end. The Macintosh seems to offer several advantages for this application, not the least of which is the inspiration provided by its well human-engineered interface. Its bitmap display is fast and can be used for presenting both text and graphics. Tools for menu building, icon construction, etc. are available in the programmer's Toolkit. We are convinced that the user deserves an interface that responds very quickly to any input, especially graphical gestures. A separate processor devoted solely to managing this part of the SDBMS-1 net seems desirable. We anticipate that the integration of this new technology into the existing MASCOT framework will be straightforward.

A more fundamental problem is how to provide the user with a model of the information structure in SDBMS-1 which will facilitate the construction of queries from a

semantic point of view. Linked to this is our desire to provide an environment that permits browsing in the database. The Entity-Relationship diagram may be a suitable way of presenting the user with the structure of the information model; we intend to consider ways of allowing the user to explore the database by navigating through a graphical representation of the ER diagram, as suggested by Zhang and Mendelzon (10). This diagram will be "intelligent" in the sense that it will know the attributes and contents of the entities, and the nature of the links between them. We envision an interface that allows the user to point at (or speak) an entity on the diagram, get the attributes associated with the entity, and find out what values occur in the table for a given attribute. By tracing a route through the diagram, and making selections from the attributes along the way, the user will be able to find data values in one entity (table) that are linked to particular values of attributes in another (e.g., to answer the question "How much ammunition for 105 mm guns does the 23rd Artillery Regiment have?"). This style of interface permits evolution of a query, rather than requiring that the user input it in a single chunk. Of course, it will be necessary to provide access to the graphical representation of data (probably through a sketch map) as well as the textual representation. We expect that the Macintosh will assist this work by providing some of the tools necessary to develop these ideas.

A future version of SDBMS must not only assist the user in query formation, it must also provide intelligent interpretation of the dialogue between itself and the user. Our research into intelligent dialogue systems (11) suggests that an important component of such systems is the model that the system holds about an individual user. A simple version of a model could be stored as a *user profile*, a file describing the user's level of expertise with the system, his preferences for default display style, and other conditions that may need to be initialized when that user signs on to the system. The interface would adapt according to the parameters specified in the user profile. A more complex and sophisticated model of the user will be needed to provide an interface that understands the context of the user's queries and that can ask for clarification or additional information needed for correct interpretation of a query. We will continue to investigate the contribution that artificial intelligence can make to this problem in the future.

Tools for the Tactical Planner

Our study of tactical planning (3) concluded that the capability to compute and display intervisibility and mobility in the terrain were desirable tools for the planner. We intend to incorporate these capabilities into SDBMS-1 via the mechanism of spatial operators. Efficient computation will require that elevation data be transformed from its current representation (as X,Y streams of contours) into a digital elevation model, a raster storage form more appropriate for the algorithms, some of which have already been developed. It will also require that we expand the data model to include other kinds of terrain information that impact on mobility and intervisibility, including soil type, gradient, vegetation characteristics, and hydrology (e.g., river current and depth). These data are not necessarily available yet in digitized form, so they will need to be acquired and entered into the database. We have developed a list of prototypical queries concerning intervisibility and mobility that we think the military planner may find useful. However, evaluation of the utility of these tools, their access through the user interface and the methods for displaying the results of computations to the user are human factors issues that will need to be resolved.

Spatial Data Representation

The representation of information in the relational model and in SDBMS-1 is object-oriented. Space is always treated as an attribute of objects rather than as the primary key upon which to organize information. For queries that are oriented directly to objects, this is satisfactory, although sometimes slow. We believe, however, that this representation may not be efficient for queries that are fundamentally concerned with the geographic or topological relation between objects. The canonical form of the question in this case is "What is here?" rather than "What is the location of this object?". We are exploring alternate representations of spatial data, including strip and quad-tree representations, hypergraphs and triangular representations. We will continue to push our use of the relational model as far as possible by mating other representations to it. Ultimately, however, it will probably be necessary to develop a new model that combines the best features of an object-oriented representation with a truly spatial one.

Time-dependent Data

A fundamental characteristic of information in a command and control environment is its dynamic nature. This has been recognized in most systems by provision for a date-time group attached to tactical information that reflects the time at which the information was updated. A more comprehensive capability to handle time-dependent data is desirable, however. It should be possible to store historical information about enemy movements or status that could be replayed in an animated form; such a presentation might provide intelligence staff with clues about the enemy's future intentions. Similarly, a commander might find it desirable to play out the future tactical consequences of an operational plan. On a more mundane level, a capability to handle time-dependent data will be necessary to allow weather predictions to be incorporated into mobility calculations, and to allow simple prediction of the current position of, for example, an enemy tank sighted a few hours earlier. Future versions of SDBMS will need to accommodate time-dependent data.

REFERENCES

1. Taylor, I., and Taylor M.M., 1983, "Psychology of Reading", Academic Press, New York, New York.
2. Bolt, R.A., 1980, Computer Graphics, 14, 262-270.
3. McCann, C., and Moogk, C., 1983, "Spatial Information in tactical planning", DCIEM Report No. 83-R-60, DCIEM, RESTRICTED.
4. Moon, G.C., Collins, S.H., and Blair, G., 1984, "Considerations for the design of an Interactive Spatial Information System (ISIS)", Contract Report 8SE80-00052, DCIEM.
5. Rhodnlus Inc., 1983, "Mistress: Relational Database Management System", Toronto, Canada.
6. Tuori, M.I., and Moon, G.C., 1984, "A topographic map conceptual data model", In Proceedings of the International Symposium on Spatial Data Handling, August 20-24, Zurich, Switzerland, Universitat Zurich-Irchel, Zurich, Switzerland.
7. McCann, C., Tuori, M., Taylor, M., Moon, G., Teasell, S., and Cote, D., 1984, "Spatial Database Management System - 1: User interface design", DCIEM Report No. 84-C-13, DCIEM.
8. Jackson, K., 1977, "Language design for modular software construction", In B. Gilchrist (Ed.), Information Processing 77, IFIP, North-Holland, Amsterdam, The Netherlands.
9. Jackson, K., and Simpson, H.R., 1975, "MASCOT - A modular approach to software operation construction and test", Technical Note 778, Royal Signals and Radar Est., Malvern, U.K.
10. Zhang, Z.-Q., Mendelzon, A.O., 1983, "A graphical query language for entity-relationship databases", In C.G. Davis, S. Jajodia, P.A. Ng and R.T. Yeh (Eds.), "Entity-relationship approach to software engineering", Elsevier Science Publishers B.V., North-Holland, Amsterdam, The Netherlands.
11. Longwoods Research Group Ltd., 1984, "Review of intelligent dialogue projects for command and control", Contract Interim Report LC687, DCIEM.



DEFENCE AND CIVIL INSTITUTE OF ENVIRONMENTAL MEDICINE
1133 Sheppard Ave West, PO Box 2000, Downsview, Ontario, Canada
Telephone (416) 635-2000